MICRO BITS

JORNAL DO CLUBE NACIONAL DOS TK/NE/SINCLAIR

ANO I Nº 2 ABRIL - MAIO 1983

Exemplar de assinante
Venda proib.!

ÍNDICE

Editorial	2
Linguagem de Máquina - Onde e Como?	3
Teste de "Benchmark" - BASIC comparada a Linguagem de Máquina	6
Como Criar um Cadastro	9
Classificados	10
Programa "Corrida de Carros"	11
Cartas	12
Lançamento da Microdigital	14
Clubes	14
Abaixo a Pirataria	15
Livros e Outras Publicações	16
Oferta aos Associados	16

==========

Na próxima edição :

- Um programa de Contabilidade de Délio Santos Lima
- Duas dicas de Bernhard Wolfgang Schön
  (1) Como evitar um LIST do seu programa em BASIC
  (2) Como consertar um programa defeituoso (com erro quando carregado da fita)
- Teclado Musical por Rui Cesar Torres
- Continuação do artigo sobre criação de um cadastro
- Última parte de "Linguagem de Máquina - Onde e Como?"
- Cartas
- Clubes
- Classificados

EDITORIAL

Ufa ! Finalmente o número 2 está saindo. A nossa intenção era de mandar cada edição aos associados no início do mês da publicação, mas aqui estamos com um mês de atraso ! Pretendemos endireitar tudo isso, mas vocês podem ficar tranquilos - o que importa é o número de edições de MICRO BITS que você recebe e que consta no seu cadastro com o Clube. Talvez seja melhor dizer que a emissão é aproximadamente mensal, por enquanto. Vocês ganham com isto, porque em vez de lançarmos um jornal bem menor, no tempo certo, estamos ampliando muito o conteúdo.

Como os artigos nossos nesta edição são mais extensos do que o normal, há menos participação de vocês, os associados, que desejaríamos fosse maior. Temos muitas cartas e programas seus para incluir, os quais terão mais ênfase a partir do Nº 3.

Um assunto, muito grave, está sendo abordado no artigo "Abaixo a Pirataria" que esperamos seja lido por vocês com muita atenção (especialmente por aqueles "piratas" que, por ventura, venham a receber uma cópia desta edição).

De vez em quando recebemos cartas cujas respostas só poderiam ser dadas individualmente, pois o assunto abordado talvez não seja do interesse dos outros associados. Embora quiséssemos responder a todos que nos escrevem, isto é impossivel. Primeiro porque não temos tempo, e segundo, cada selo, envelope, papel custa dinheiro, e teríamos que aumentar o valor da assinatura para cobrir estas despesas adicionais. Mas não desistam de nos escrever. Ao contrário, precisamos de suas sugestões, programas, etc., para incluirmos no seu jornal.

Não temos vinculação com nenhum fabricante de microcomputadores ou outra entidade, então vocês podem ter certeza que qualquer análise de equipamento ou software que fizermos será sincera.

Também não temos propaganda que deixa o jornal mais grosso e aumenta sensivelmente a renda, nem desenhos desnecessários que ocupam espaço que pode ser melhor utilizado.

O nosso apelo publicado no Micro Bits Nº 1 ainda está valendo - precisamos a sua ajuda para conseguirmos um número suficiente de associados para que possamos manter o jornal circulando e fortalecer o Clube. Como o custo da impressão na gráfica baixa (por cada cópia) com o aumento do número de cópias feitas, só poderemos continuar com um jornal deste tamanho com muito mais associados do que possuimos atualmente. Do contrário, teremos que reduzir drasticamente o número de folhas em cada edição.

Assim esperamos que vocês mostrem o jornal e divulguem o Clube para seus colegas para que possamos aumentar o número de associados.

= = = = = = = = = = = = = = = = = = =

CORRESPONDÊNCIA

Mandem as suas cartas para :-

David Anderson
MICRO BITS
CLUBE NACIONAL DOS TK/NE/SINCLAIR
Caixa Postal 12.464
04798 - SÃO PAULO - SP

= = = = = = = = = = = = = = = = = = =

Todos os direitos de reprodução de MICRO BITS estão reservados, e qualquer reprodução com fins comerciais ou didáticos só poderá ser feita mediante autorização prévia.
Os artigos e cartas assinados são da responsibildade dos autores, as opiniões não necessariamente coincidindo com as desta publicação.

LINGUAGEM DE MÁQUINA - ONDE E COMO?

O que é linguagem de máquina ?

Linguagem de máquina é a linguagem fundamental do computador. As outras - BASIC, FORTRAN, COBOL, PASCAL, etc. - são chamadas linguagens de alto nível, que, para serem executadas num computador precisam ser "traduzidas" para a linguagem de máquina.

Esta tradução é feita através de :-

A: Um compilador - onde o programa fonte (em FORTRAN, por exemplo) é traduzido num programa objeto (em linguagem de máquina) antes que este possa ser executado.

B: Um interpretador - cada linha do programa fonte é interpretada para linguagem de máquina e executada em seguida (sem passar por um programa objeto intermediário).

A grande diferença entre um compilador e um interpretador é que o 2º interpreta cada linha todas as vezes que é executada. No caso de uma linha estar dentro de um loop "FOR... NEXT" (por exemplo) e executada 1000 vezes, a linha é interpretada 1000 vezes, enquanto no 1º caso um programa compilado já existe em linguagem de máquina (a tradução foi feita uma vez só) e, consequentemente, é executado muito mais rapidamente.

Existe uma outra linguagem intermediária, de baixo nível, chamada ASSEMBLER, que é geralmente usada para escrever o software básico de um computador - ou seja os sistemas operacionais (monitores) e compiladores, embora os dois possam ser escritos numa linguagem de alto nível.

O ASSEMBLER é necessário,porque programar diretamente em linguagem de máquina é difícil - na programação, correção, modificação e entendimento do que o programa faz.

Para dar um exemplo, aqui está um programa bem simples, escrito em linguagem de máquina, que pode ser rodado em nossos micros :-

```
ØØ11111ØØØ1ØØØØØ
11ØØØ11ØØØØ1ØØØØ
ØØØØØ11ØØØØØØØØØ
Ø1ØØ1111
11ØØ1ØØ1
```

É óbvio o que este programa faz e .....

Como ? Você não entendeu nada ???

Talvez reescrevendo-o no sistema hexadecimal fique mais claro. É -

```
3E2Ø
C61Ø
Ø6ØØ
4F
C9
```

Bem, é mais fácil ler, mas ainda não dá para entender o que faz. Em ASSEMBLER,o programa é isto -

```
LD  A,32
ADD A,16
LD  B,Ø
LD  C,A
RET
```

Para quem não conhece ASSEMBLER , isto não significa muito, mas podemos comparar cada linha deste programa com o equivalente em BASIC. Basta dizer que a primeira linha acima é igual a :

```
LET A = 32
```

e ADD A,16 em BASIC seria :

```
LET A = A + 16
```

A última linha, RET, tem o equivalente em BASIC que é quase igual. É o RETURN.Assim podemos reescrever o nosso programa em BASIC :-

```
LET A = 32
LET A = A + 16
LET B = Ø
LET C = A
RETURN
```

e agora fica bem mais claro o que vem a ser este programa (ou subrotina, que seria mais certo). Supondo que o programa em linguagem de máquina foi carregado no RAM do micro a partir do endereço - 3ØØØØ (que será explicado depois) o resultado do comando direto(ou utilizado dentro de um programa) de :

```
PRINT USR 3ØØØØ
```

seria o aparecimento do valor "48" na tela do seu TV, correspondendo ao valor de -

256*B + C

Ou 256*Ø + 48 = 48

Não vou elaborar mais, por enquanto, os fundamentos de linguagem de máquina, ou de ASSEMBLER. Voltarei ao assunto em futuras edições deste jornal. O propósito desta introdução é de mostrar como, e onde, podem ser carregados os programas no microcomputador.

Onde carregar programas em linguagem de máquina ?

Temos três maneiras possíveis de carregar um programa em linguagem de máquina, cada uma com suas vantagens e desvantagens. Vou descrever as três e depois fazer uma comparação entre elas.

A - O Uso de "REM"

Talvez seja o mais fácil para implementar, e também para arquivar em fita, embora este método crie um "comentário" bem ininteligível . Dúvidas com relação a este ponto esclarecerei mais adiante.

Para se utilizar este método, seu programa (escrito em BASIC) deve começar com uma linha "REM" contendo um número de caracteres igual ou maior do que o programa que você está pretendendo escrever em linguagem de máquina. Não importa quais caracteres, só quantos. Uma maneira de se ter certeza que tem a quantidade certa é entrar os caracteres 1 á Ø, e repeti-los quantas vezes forem necessárias. Assim -

```
  1 REM 123456789Ø123456789Ø123
456789Ø123456789Ø
```

reserve espaço para um programa de até 4Ø bytes de comprimento (cada carácter ocupa 1 byte na memória).

Antes de começar digitar nosso programa em linguagem de máquina, entre com este em BASIC :-

```
  1Ø REM 123456789Ø123456789Ø123
456789Ø123456789Ø
  2Ø LET INI = 165Ø9
  3Ø FOR I = INI TO INI + 2Ø
  4Ø LET V = PEEK I
  5Ø PRINT I;TAB 7;V;TAB 12;CHR$
 V;TAB 18;CHR$ (INT (V/16)+28);C
HR$ (V-INT (V/16)*16+28)
  6Ø NEXT I
  7Ø STOP
```

Agora digite RUN. Na sua tela aparecerá :

165Ø9	Ø		ØØ
1651Ø	1Ø	[ENT]	ØA
16511	42	E	2A
16512	Ø		ØØ
16513	234	REM	EA
16514	29	1	1D
16515	3Ø	2	1E
16516	31	3	1F
16517	32	4	2Ø
16518	33	5	21
:			
:			
:			
16529	34	6	22

A primeira coluna é o endereço (em decimal); a segunda o código do conteúdo deste endereço; a terceira o carácter equivalente, e o quarto o código em hexadecimal.

Por exemplo, endereço 16513 contém o valor 234 em decimal, o carácter equivalente sendo REM (o "REM" da linha número 1Ø), cujo código hexadecimal é EA.

Assim estamos vendo parte da primeira linha do programa, que sempre começa com o endereço 165Ø9. Os endereços 165Ø9 e 1651Ø contém o número da linha (Ø*256 + 1Ø = 1Ø) com o byte mais significativo primeiro. 16511 e 16512 contém o comprimento da linha (contando a partir do REM, que ocupa um byte, até o último carácter, mais o NEW LINE) que é 42 + Ø*256 = 42, com o byte menos significativo sendo primeiro. Assim temos 4Ø caracteres numéricos, mais o REM, mais o NEW LINE = 42. A partir de 16513 temos o "REM", seguido pelos caracteres "1","2","3", ...

Vamos continuar o nosso programa , modificando-o como é mostrado abaixo -

```
  7Ø INPUT P$
  8Ø CLS
  9Ø PRINT "DIGITE EM HEX"
 1ØØ LET END = 16514
 11Ø LET S$ = ""
 12Ø IF S$ = "" THEN INPUT S$
 13Ø IF CODE S$<28 OR CODE S$>43
 THEN STOP
 14Ø POKE END,16*(CODE S$-28) + CO
DE S$(2)-28
```

```
15Ø LET END = END + 1
16Ø LET S$ = S$(3 TO)
17Ø GOTO 12Ø
1ØØØ REM 123456789Ø1234567890123
456789Ø123456789Ø
```

Agora você já pode digitar um programa em linguagem de máquina e rodá-lo. Digitando RUN novamente, aparecerá a mesma informação como antes. Depois de digitar um NEW LINE (em resposta a linha 7Ø acima) a mensagem "DIGITE EM HEX" aparecerá na tela. Então vamos digitar o nosso primeiro programa, usando a versão hexadecimal. Você pode digitar tudo numa linha só :

"3E2ØC61ØØ6ØØ4FC9"

Ou exatamente como está escrito :

"3E2Ø" e depois
"C61Ø"
"Ø6ØØ"
"4F"
"C9"

ou qualquer combinação de caracteres, na mesma sequência, só que o número de caracteres em cada linha digitada deverá ser par. Para terminar, digite um carácter fora daqueles usados no sistema hexadecimal (por exemplo, use uma letra na faixa de G até Z).

Você tem, agora, o nosso programa em linguagem de máquina, no início do programa em BASIC, embutido na linha 1Ø.

Vamos listar o programa novamente . Digite LIST. Aparecerão alguns caracteres estranhos na primeira linha, que era -

```
1Ø REM 123456789Ø12 ....
```

A linha foi modificada para -

```
1Ø REM Y4LEN (■ ?TAN 9Ø1234567
89Ø123456789Ø1234567890
```

Digitando RUN veremos -

165Ø9	Ø		ØØ
1651Ø	1Ø	■■	ØA
16511	42	E	2A
16512	Ø		ØØ
16513	234	REM	EA
16514	62	Y	3E
16515	32	4	2Ø
16516	198	LEN	C6
16517	16	(	1Ø
16518	6	■	Ø6
16519	Ø		ØØ
1652Ø	79	?	4F
16521	2Ø1	TAN	C9
16522	37	9	25
16523	28	Ø	1C

:
:
etc.

Ahá ..... !!!! Quem estava tentando decifrar os "REM"s no início de alguns programas comerciais, mas sem sucesso, talvez agora comece a compreender. O programa que digitamos depois da mensagem "DIGITE EM HEX" foi gravado na memória nos endereços 16514 a 16521 (ver a última coluna acima, onde apareceram os mesmos códigos que digitamos). E quando listamos o programa (com LIST) a linha 1Ø tem os caracteres 12345678 trocados por Y,4,LEN,(,■,□,? e TAN (onde □ = "SPACE"), correspondendo aos códigos hexadecimais 3E, 2Ø, ... , C9. Então aqueles símbolos estranhos são justamente o nosso programa que acabamos de digitar. Como o monitor no ROM não sabe que isto é um programa em linguagem de máquina, mas está esperando um programa em BASIC, ele traduz o conteúdo do "REM" nos caracteres equivalentes que tem no seu "conjunto de caracteres" (ver o apêndice correspondente no seu manual de proprietário).

Para rodar o novo programinha, precisamos sair do programa em BASIC, que está esperando você entrar NEW LINE. Para isso há duas maneiras - ou digite RUBOUT (SHIFT/Ø) e depois STOP, ou digite NEW LINE e responda a "DIGITE EM HEX"com uma letra entre G e Z. Agora faça o comando direto "PRINT USR 16514", e aparecerá o valor "48" na tela.

- O que significa a linha 1ØØØ?

Esta é uma cópia exata da linha 1Ø original (mas com o número da linha sendo 1ØØØ). Se você quiser gravar o programa em BASIC em fita, mas sem o programa em linguagem de máquina, digite "LIST 1ØØØ", e depois SHIFT/EDIT, e retire dois zeros do número da linha (SHIFT/RUBOUT duas vezes). Após um NEW LINE, a linha 1Ø aparecerá no início do programa na sua forma original.

Você vai precisar deste programa para rodar os testes de "Benchmark" nesta edição do jornal, e assim deverá gravar em fita usando - SAVE "HEXTST".

Na próxima edição do jornal continuarei com a descrição dos outros dois métodos para se armazenar programas em linguagem de máquina no seu micro.

- - - : - - -

TESTE DE "BENCHMARK"
BASIC COMPARADA A LINGUAGEM DE MÁQUINA

Programar em linguagem de máquina é difícil e muito mais demorado que numa linguagem de alto nível. Então porque desenvolver programas nesta linguagem para os nossos micros?

Posso citar algumas razões -

1. Um programa em linguagem de máquina roda muito mais rapidamente que um desenvolvido em BASIC.
2. Um programa em BASIC ocupa muito mais espaço na memória do computador.
3. "Pirataria" é mais difícil. Primeiro, porque podemos "desligar" a tecla "BREAK" (ou qualquer outra tecla) que dificultará a listagem do programa e, consequentemente, "disfarçá-lo" para parecer diferente do original.

Neste artigo vou mostrar quanto mais rápido roda um programa em linguagem de máquina comparando com o equivalente em BASIC. Os programas utilizados serão descritos, e você pode rodá-los para cronometrar e comparar com meus resultados, ou fazer seus próprios testes de "Benchmark".

Os testes foram feitos tanto em 'SLOW' como em 'FAST', e os resultados também analisados em seguida.

Foram escolhidos três programas bem simples, usando loops 'FOR ... NEXT'. Obviamente os resultados dos testes devem modificar-se no caso de utilizar outros programas e, provavalmente, alguem poderia sugerir outros programas em linguagem de máquina, menores, mais rápidos, etc., que mudariam os resultados. Mas o intuito deste artigo é o de mostrar a ordem de grandeza da diferença entre as duas linguagens, e não como programar eficientemente, nem instruir como programar em linguagem de máquina.

Os Programas de Teste

Os três programas em BASIC estão da forma :-

```
2Ø PRINT "INICIO"
3Ø FAST       (ou SLOW)
4Ø ....
   ....   programa
7Ø PRINT valor
8Ø SLOW
```

e os três em linguagem de máquina tem um programa padrão BASIC para chamá-los -

```
2Ø PRINT "INICIO"
3Ø FAST     (ou SLOW)
4Ø PRINT USR 16514
5Ø SLOW
6Ø STOP
```

A cronometragem dos testes foi feita assim :-

1. Digitando 'RUN', começa a cronometragem logo que aparece a mensagem "INICIO".
2. Ao aparecer o valor numérico na tela, termina-se a cronometragem.
3. Cada teste foi feito usando-se os seguintes loops :
   (i) De Ø a 16383 (FAST/SLOW)
   (ii) De Ø a 32767 (FAST/SLOW)
   (iii) De Ø a 65535 (FAST/SLOW)

Para distinguir os três pares de programas de teste, eles estão identificados assim :-

Programas B1 e M1 - BASIC 1 e linguagem de máquina 1
Programas B2 e M2 - BASIC 2 e linguagem de máquina 2
Programas B3 e M3 - BASIC 3 e linguagem de máquina 3

Programa B1

```
1Ø LET L = -1
2Ø PRINT "INICIO"
3Ø FAST     (ou SLOW)
4Ø FOR I = Ø TO 16383
5Ø LET K = K+1
6Ø NEXT I
7Ø PRINT K
8Ø SLOW
```

(Troque 16383 para 32767 e 65535 para os outros testes)

Programa M1

Para digitar este programa carregue o "HEXTST" no seu micro, apague as linhas 2Ø a 7Ø, e inclua as seguintes linhas :

```
2Ø PRINT "INICIO"
3Ø FAST    (ou SLOW)
4Ø PRINT USR 16514
5Ø SLOW
6Ø STOP
```

Digite RUN 8Ø e depois os códigos hexadecimais na coluna esquerda abaixo. As colunas da direita contém o programa em ASSEMBLER :

```
3E3F   (*)          LD  A,63
Ø1ØØØØ              LD  BC,Ø
11ØØØØ              LD  DE,Ø
13           LOOP1  INC DE
Ø3                  INC BC
BA                  CP  D
2ØFB                JRNZ LOOP1
3EFF                LD  A,255
13           LOOP2  INC DE
Ø3                  INC BC
BB                  CP  E
2ØFB                JRNZ LOOP2
C9                  RET
```

(*) 3F corresponde ao loop Ø a 16383. Para os outros dois, troque para:-

(a) Ø a 32767 - 7F
(b) Ø a 65535 - FF

Para rodar o teste - digite RUN, comece a cronometragem ao aparecer "INICIO" e termine quando o valor número for mostrado.

Para mudar o 3E3F para 3E7F (ou 3EFF) para os outros testes, digite RUN 8Ø, e em seguida "3E7FZ" (ou "3EFFZ") depois de "DIGITE EM HEX".

Programa B2

```
1Ø LET K = -1
2Ø PRINT "INICIO"
3Ø FAST       (ou SLOW)
4Ø FOR I = 16383 TO Ø STEP -1
5Ø LET K = K+1
6Ø NEXT I
7Ø PRINT K
8Ø SLOW
```

(Troque 16383 para 32767 e 65535 para os outros testes)

Programa M2

Carregue da mesma forma que para M1.

```
3EØØ                LD  A,Ø
Ø1ØØØØ              LD  BC,Ø
11FF3F   (*)        LD  DE,16383
1B           LOOP1  DEC DE
Ø3                  INC BC
BA                  CP  D
2ØFB                JRNZ LOOP1
1B           LOOP2  DEC DE
Ø3                  INC BC
BB                  CP  E
2ØFB                JRNZ LOOP2
C9                  RET
```

(*) FF3F corresponde a 16383
FF7F " " 32767
FFFF " " 65535

Programa B3

```
1Ø PRINT "INICIO"
2Ø FAST       (ou SLOW)
3Ø FOR I = Ø TO 16383
4Ø NEXT I
5Ø PRINT I
6Ø SLOW
```

Programa M3

```
21FF3F   (*)        LD  HL,16383
7C                  LD  A,H
Ø1ØØØØ              LD  BC,Ø
Ø3           LOOP1  INC BC
B8                  CP  B
2ØFC                JRNZ LOOP1
7D                  LD  A,L
Ø3           LOOP2  INC BC
B9                  CP  C
```

```
2ØFC            JRNZ LOOP2
C9              RET
```

(*) Troque para 21FF7F e 21FFFF como acima em M1.

Sequência dos Testes

Digitando um programa por vez, foram feitos os seis testes em BASIC (três loops, cada um em FAST e depois em SLOW), repetindo todos para verificar as temporizações. Assim, começando com programa B1 -

```
1º teste    3Ø FAST
            4Ø FOR I = Ø TO 16383

2º teste    3Ø SLOW
            4Ø FOR I = Ø TO 16383

3º teste    3Ø FAST
            4Ø FOR I = Ø TO 32767
   .
   .
   .
   etc.
```

repetindo para os programas B2 e B3, e finalmente rodando os 6 testes equivalentes em linguagem de máquina.

Resultados

Os tempos mostrados na tabela abaixo são em minutos, segundos, e decimais de segundos no formato mm:ss,d. Tempos seguidos por um ponto de interrogação ("?") são duvidosos por serem de tão curta duração.

		BASIC		L. DE MÁQUINA	
Série	Loop	FAST	SLOW	FAST	SLOW
1	Ø a 16383	2:Ø1,7	12:52,7	Ø:ØØ,1?	Ø:Ø1,6
(B1 e	Ø a 32767	4:Ø4,5	25:52,1	Ø:ØØ,3?	Ø:Ø2,5
M1)	Ø a 65535	8:11,2	51:58,2	Ø:ØØ,6?	Ø:Ø4,3
2	Ø a 16383	1:59,8	12:4Ø,8	Ø:ØØ,2?	Ø:Ø1,6
(B2 e	Ø a 32767	4:ØØ,7	25:28,5	Ø:ØØ,3?	Ø:Ø2,6
M2)	Ø a 65535	8:Ø3,6	51:11,Ø	Ø:ØØ,7?	Ø:Ø4,3
3	Ø a 16383	1:13,3	7:44,8	Ø:ØØ,2?	Ø:Ø1,5
(B3 e	Ø a 32767	2:26,3	15:32,8	Ø:ØØ,3?	Ø:Ø2,2
M3)	Ø a 65535	4:54,8	31:11,2	Ø:ØØ,5?	Ø:Ø3,6

Antes de se poder comparar os valores, é necessário subtrair o tempo que não consta propriamente do cálculo da temporização do loop . Por exemplo, considerando o programa -

```
2Ø PRINT "INICIO"
3Ø SLOW      (ou FAST)
4Ø PRINT USR 16514
5Ø SLOW
6Ø STOP
```

devemos descontar o tempo para processar o SLOW da linha 3Ø e o PRINT da linha 4Ø. O loop do programa de máquina corresponde ao "USR 16514".

Mas isto é fácil calcular. Supondo que este tempo (que é constante, independendo do número de loops do programa de linguagem de máquina) seja C, e o tempo para fazer os loops de Ø a 16383 seja N. Usando os valores de M3 (SLOW) acima, posso escrever :-

```
C + N  =  1,5      . . . . . .(i)
C + 2N =  2,2      . . . . . .(ii)
C + 4N =  3,6      . . . . . .(iii)
```

Subtraindo (i) de (ii) temos -

```
C - C + 2N - N = 2,2 - 1,5
ou  N = Ø,7
```

Subtraindo (ii) de (iii) -

```
3N = 2,1
ou  N = Ø,7
```

e subtraindo (ii) de (iii)

```
2N = 1,4
ou  N = Ø,7
```

Assim os tempos verdadeiros para completar os loops do programa em linguagem de máquina são:

```
(i)    Ø a 16383  =  Ø,7 segundos
(ii)   Ø a 32767  =  1,4 segundos
(iii)  Ø a 65535  =  2,8 segundos
```

Por incrível que pareça, o tempo gasto nas linhas 3Ø e 4Ø ("PRINT"), que é 1,6 - Ø,7 = Ø,9 segundos, é maior do que o gasto pelo loop

de Ø a 16383!

No caso dos programas B1,B2 e B3 , houve uma surpresa. Usando o mesmo sistema de cálculo para retirar o tempo das linhas não dentro dos loops, o constante C é negativo! O que parece que está acontencendo é que o tempo de um loop aumenta quando o contador do loop aumenta, e não é constante. Assim quando I = Ø (em B1, por exemplo) o tempo para completar um loop é menor do que quando I = 65535.

Recalculando os tempos da tabela anterior (somente para os programas em linguagem de máquina e SLOW) subtraindo os tempos que não correspondem aos loops, temos o seguinte -

	M1	M2	M3
Ø a 16383	Ø,9	Ø,9	Ø,7
Ø a 32767	1,8	1,8	1,4
Ø a 65535	3,6	3,6	2,8

Resultados

Calculei as razões dos tempos - BASIC/Linguagem de máquina,e FAST/SLOW para BASIC, para os três grupos de programas. Aqui estão -

A: Razão dos tempos BASIC / Linguagem de máquina

	Ø a 16383	Ø a 32767	Ø a 65535
B1 / M1	858	862	866
B2 / M2	845	849	853
B3 / M3	664	666	668

B: Razão dos tempos SLOW / FAST

	Ø a 16383	Ø a 32767	Ø a 65535
B1	6,35	6,35	6,35
B2	6,35	6,35	6,35
B3	6,34	6,38	6,35

Conclusões

Podemos concluir, baseados nestas últimas duas tabelas, que um programa em linguagem de máquina pode rodar até aproximadamente 900 vezes mais rápido do que o equivalente em BASIC.Mas devo repetir que isto depende muito dos programas estudados e só pode ser considerada uma aproximação grosseira.

A razão de se ter uma diferença tão grande entre as duas linguagens vem, em parte, do fato que não foram utilizadas as rotinas da 'Calculadora' do ROM, mas as instruções básicas do Z80 (aritmética inteira, e não ponto flutuante). O valor normalmente indicado para esta razão BASIC/L. Máquina é entre 30 e 40 para programas equivalentes.

Agora,as razões de tempo dos programas em BASIC rodados em SLOW / FAST mostraram pouca variação.

- Podemos concluir que esta razão de 6,35 é constante, e independe dos programas examinados ? Se você quiser saber a resposta, faça suas próprias pesquisas!

- - - : - - -

COMO CRIAR UM CADASTRO

INTRODUÇÃO

Neste artigo, que será divulgado em várias partes, vou mostrar como você pode criar um cadastro no seu micro, modificando-o conforme suas necessidades.

A informação contida no cadastro pode ser de qualquer natureza, mas aqui estão algumas sugestões para seu uso :-

1. Uma lista telefônica, contendo nomes, endereços e números de telefone. Pode ser utilizada para procurar o número do telefone de uma determinada pessoa, ou listar os nomes, etc., de pessoas morando em certa cidade ou estado, relacionar as pessoas cujos telefones começam com 257 etc.
2. Uma relação de clientes, com dados pessoais,limite de crédito e quanto deve.
3. Uma listagem de sua coleção de selos com data e preço de compra, valor atual e código de referência de catálogo.
4. Um controle pessoal da sua conta bancária.
5. Uma agenda eletrônica mostrando futuros compromissos.

Como pode ver, as aplicações são diversas, e provavalmente você tem um "hobby" ou outra atividade que beneficiaria com uso de um cadastro.

Vou descrever um programa para criar um cadastro de associados do nosso Clube, e mostrar como os dados podem ser verificados, corrigidos, ordenados e listados. A partir deste programa você pode criar um para seu uso, fazendo as modificações necessárias.

Para rodar este programa, será

necessário um mínimo de 16K de RAM - se houver mais, melhor ainda ! Devo avisar que o programa em si ocupa muita memória no micro - há rotinas para entrada de dados, correção e eliminação, listar, ordenar, arquivar em fita, verificação de valores números, e para imprimir mensagems de erro na tela. Talvez não seja necessário um programa tão extenso, mas a idéia é para mostrar o que pode ser feito, e você escolhe o que o acha suficiente para seus requisitos. Obviamente, quanto menor o programa, maior o número de dados que podem ser armazenados.

Uma das deficiências de nosso micros é o fato que não podermos separar os dados do programa, e para arquivar, temos que gravar os dois (programa junto com os dados) na fita cassette. Isto significa que não podemos escrever um programa só para digitar novos dados, outro programa para corrigir a informação contida, outro para listar , etc. Não, precisamos criar um programa único contendo tudo que vamos precisar. O problema que surge é que qualquer modificação feita no programa, futuramente pode resultar em memória insuficiente para continuar. Imagine, depois de digitar 100 nomes e endereços, tentamos colocar uma rotina pequena para verificar os CEPs dos endereços, mas chegamos ao limite de memória! Neste caso teríamos 3 opções :

(i) Ficar com o programa velho, sem a nova subrotina.

(ii) Modificar o programa, retirando comentários, linhas desnecessárias, etc.

(iii) Apagar todos os dados, incluir a nova subrotina, e começar digitar tudo novamente (neste caso com 90 nomes no arquivo, em vez de 100).

Para não ter que tomar uma decisão destas, há duas alternativas :

(i) Terminar o programa completamente, colocando todas as subrotinas necessárias, antes de dimensionar e digitar os dados.

(ii) Deixar espaço suficiente para futura expansão.

Prefiro tomar um caminho intermediário entre as duas opções - desenvolver e depois digitar o programa completo, mas também deixando um pouco de espaço para incluir pequenas modificações. A inclusão de linhas de comentários, de vez em quando, é útil. Quando se necessita de mais memória, elas podem ser eliminadas sem nenhum problema.

Outra deficiência nos micros é o fato de que todos os dados contidos na mémoria são apagados depois de digitar RUN. Se você tivesse gasto duas horas digitando dados no teclado e, antes de gravar a primeira cópia em fita, você se distrai e digita RUN .... - tudo seria perdido !

Uma maneira de evitar isto seria um lembrete para usar GOTO 5ØØ (por exemplo) cada vez que se quer rodar o programa. Mas isto pode ser esquecido facilmente. Felizmente há outra alternativa através do uso do comando SAVE, que será explicado detalhadamente adiante. Depois de gravar seu programa (com os dados) em cassette, a próxima vez que carregar no micro (LOAD) o programa começará a rodar automaticamente, sem nenhuma ação sua. Assim não precisará lembrar de digitar GOTO ... ao invés de RUN.

Posso afirmar que a linha contendo SAVE é a mais importante do programa inteiro !

Este artigo será continuado na próxima edição de MICRO BITS.

. . . : . . .

CLASSIFICADOS

VENDO CP-200 por Cr$170.000,00 e uma calculadora SHARP 1211 com impressora e interface para K-7 por Cr$160.000,00 (ambos com garantia) - J. Maldonado, Tel: (011) 452-6617 ou 452-6835.

TROCO, compro e/ou vendo programas para NEZ8000. André Zielasko, R. Emancipação 402, TRAMANDAÍ, CEP 95590 - RS.

VENDE-SE livro "The Alien, Numbereater and other Programs for Personal Computers" por John Race. Como escrever programas de jogos, simulações, etc. Para o PET 2001, 8K. Cr$6.000,00. Caixa Postal - 12.464, 04798 - SÃO PAULO - SP.

PROCURAM-SE programas em linguagem de máquina de alta qualidade para comercialização. NÃO queremos cópias. Envie detalhes sobre o(s) programa(s) e os recursos necessários (SLOW, 16K, etc.) para Caixa Postal 12.464, 04798 - SÃO PAULO - SP.

PROGRAMA "CORRIDA DE CARROS"

Este programa mostra de 2 até 7 carros na tela (o número de carros sendo selecionado ao rodar o programa) e cada carro move na direção da linha de chegada até um carro chega primeiro.

Para rodar o programa, digite RUN, e responda a pergunta "QUANTOS CARROS" com um valor entre 2 e 7 (inclusive). Os carros aparecem na tela e começam a andar depois de digitar NEW LINE.

O programa só roda em SLOW, e pode ser executado em micros de 1K de RAM para cima.

```
  1Ø REM CORRIDA DE CARRO
  2Ø REM MICRO BITS MARCO/83
  3Ø PRINT AT Ø,Ø;"QUANTOS CARROS?(2 A
7)";
  4Ø INPUT DM
  5Ø IF DM<2 OR DM>7 THEN GOTO 3Ø
  6Ø PRINT DM
  7Ø DIM P(DM)
  8Ø LET PRIM = Ø
  9Ø PRINT "NEW LINE P/COMECAR"
 1ØØ INPUT I$
 11Ø CLS
 12Ø FOR T = 1 TO DM
 13Ø LET Y = 3*(T-1)
 14Ø IF RND>.5 AND PRIM = 1 THEN
GOTO 23Ø
 15Ø LET P(T) = P(T) + 1
 16Ø PRINT AT Y,P(T); "□□■■■"
 17Ø PRINT AT Y+1,P(T);"□■■■■■
■";TAB 31;"■"
 18Ø PRINT AT Y+2,P(T);"□0□□□□0"
 19Ø PRINT AT Y+1,P(T)+3;T
 2ØØ IF P(T)<26 THEN GOTO 23Ø
 21Ø PRINT AT Y+1,1;T;" GANHA"
 22Ø GOTO 7Ø
 23Ø NEXT T
 24Ø LET PRIM = 1
 25Ø GOTO 12Ø
```

As variáveis usadas são :-

DM	Nº de carros selecionados - usado para dimensionar a matriz P.
P(DM)	Posição de cada carro (na horizontal)
Y	Posição de cada carro (na vertical)
PRIM	Inicializado para Ø. Força o desenho de todos os carros no início - linha 14Ø.
T	Variável de controle do loop FOR ...NEXT

Resumo do programa

linhas 3Ø a 9Ø — Inicialização. A matriz P é dimensionada para DM carros.

linha 1ØØ — Um NEW LINE começa a corrida.

linhas 12Ø a 23Ø — Este loop é repetido até que um dos carros chegue ao final.

linha 13Ø — Define a posição do carro.

linha 14Ø — A primeira vez que o loop é executado (para todos os carros), cada carro aparece na tela. Isto é forçado porque PRIM ainda é zero, e a condição de RND>.5 AND PRIM = 1 não é verdade e "GOTO 23Ø" não pode ser executado. (Para entender o significado de PRIM, retire a condição "AND PRIM = 1")
Depois de completar o loop uma vez para todos os carros, PRIM muda para 1, e assim só depende no valor de RND. Se RND for<=.5 então o carro definido por T avança uma posição. Se for>.5 o carro fica parado.

linhas 15Ø a 18Ø — Desenha o carro na tela. Para maior clareza, os símbolos gráficos das linhas 16Ø a 18Ø são especificados em baixo :

16Ø = "SP,SP,g8,g7,g5"
17Ø = "SP,gSP,gSP,gSP,gSP,gSP,gSP" e "g4"
18Ø = "SP,0,SP,SP,SP,SP,0"

onde :

SP - significa espaço (SPACE)
g5 - significa o símbolo gráfico na tecla "5" (conseguido através de SHIFT 5 em modo gráfico.)
g4,g7,g8 - ver para g5
gSP espaço inverso ("GRAPHIC SPACE")

(as vírgulas são somente para separar os caracteres e não devem ser digitadas).

linha 19Ø — Coloca o número no carro.

linha 2ØØ — Se o carro ainda não chegou ao final, então continue.

linhas 21Ø e 22Ø — Quando um carro chega ao final, imprime o número dele na tela e volta para linha 7Ø para começar outra corrida (com o mesmo número de participantes).

linha 25Ø — Repete o loop 12Ø a 23Ø.

- Porque as linhas 16Ø a 18Ø tem um espaço (ao mínimo) no início do "string"? Tire o espaço para ver o que acontece !

- - - : - - -

CARTAS ... CARTAS ... CARTAS ... CARTAS ... CARTAS

Amigo David,

Em primeiro lugar, permita-nos chamá-lo de "amigo" já que assim o consideramos e pretendemos fazer parte do CLUBE NACIONAL DOS TK/ NE/ SINCLAIR, o que nos tornará uma roda de amigos ao redor dos micros.

Recebemos (e agradecemos) o 1º número do MICRO BITS, que muito nos agradou, e já estamos enviando nossas fichas de assinaturas,acompanhadas dos respectivos cheques nominais a você. Estamos"xerocando"e distribuindo as referidas fichas entre outros colegas que se interessaram. Já germina, por aqui, sua sugestão de fazermos reuniões e até um "sub-clube" congregando interessados no assunto.

Na leitura e aplicação do MICRO BITS Nº 1, observamos o seguinte:

- carta nº 2, do Roberto Carlos Grings, (de Passo Fundo tchê!), rodamos seu programa de tiro ao alvo mas não conseguimos - em nenhum instante - o acerto do alvo.Será que não houve algum erro tipográfico? , eis que revisamos bem nossa listagem. Salientamos que não temos - ainda - condições de analisar o programa (pois somos principiantes)e ver se dá para acertá-lo.
- carta nº 1, do Luiz Carlos Nardy : ele informa ter elaborado uma espécie de VU para facilitar o LOAD. Nosso maior problema é justamente esse, a "carga" do gravador para o micro. Será que o Nardy poderia publicar seu trabalho ou o Clube providenciar contatos para - se fôr o caso - adquirirmos seu "know-how"?
-Em se falando dessa dificuldade, que deve ser comum a todos, alguém sabe quando a Microdigital lançará no mercado o diskette para o TK82-C?

Ah, consta de nossa ficha mas é bom lembrar por aqui. Temos, ambos, TK82Cs, com expansão para 16K.

David, por enquanto é só.Aguardamos novos números e nos colocamos ao inteiro dispor de todos vocês para qualquer assunto de por estas bandas de fronteira (somos cidade gêmea com Rivera, no Uruguai).

Um abraço e byby

RICARDO FERNANDES CONDE e JÚLIO CEZAR MASSA RODRIGUEZ, Santana do Livramento, RS.

. . . . .

µB: Modifique o programa "tiro ao alvo" de Roberto Carlos Grings, incluindo a linha :

135 PRINT AT 1Ø,X;T

que dará o número necessário para acertar o alvo. Caso não funcione, você deve ter cometido algum erro ao copiar o programa - verifique novamente . Utilizei esta linha, também modificando a temporização na linha 14Ø para "PAUSE 3ØØ" para que minha filha Helena, de três anos, pudesse jogar. Ela gostou!

Pedimos que Luiz Carlos Nardy nos mande detalhes sobre o seu VU, e também a adaptação SLOW para o NEZ8ØØØ. Caso vocês, ou outros leitores, queiram corresponder diretamente com ele,escrevam para :-

LUIZ CARLOS NARDY
Caixa Postal 28
17200 - JAÚ - SP

- - - : - - -

Caro David,

A criação do Clube Nacional dos TK/ NE/Sinclair veio suprir uma lacuna, a meu ver inadmissível, nas revistas de microinformática, que ainda não viram a importância que tem a grande quantidade de usuários deste tipo de equipamento, em que vejam atendidos de forma clara e rápida seus problemas específicos (de software, hardware,manutenção, intercâmbio, etc).

O centro destes problemas se situa na atuação da Microdigital, principalmente na área de pós-venda.

Seus programas são: falhos, mal gravados, de difícil acesso, pouca divulgação.

Os periféricos são: pouco encontráveis no mercado(joystick), com problema mecânico no tracionador (printer), com encaixe falho (RAM).

O seu atendimento é: por revendedor é falho, seu próprio atendimento é nulo;refletindo assim falta de adequado treinamento em todos os níveis.

Os primeiros compradores (como eu em dezembro de 1981) foram punidos inicialmente no preço (noventa mil cruzeiros), com o processador sem SLOW, péssimo estojo, embalagem ruim, manual fotocopiado (a parte sobre Assembler é ilegível).

A empresa foi supreendida pelo grande sucesso de seu equipamento, reagindo muito devagar na sua relação com os usuários (prometendo o que não pode logo cumprir).

Os usuários do TK, que já tem razoável conhecimento de BASIC, necessitam ter agora uma divulgação completa e detalhada de Assembler (principalmente como montar seus códigos de máquina). É importante um curso de como montar jogos,

como aprofundamento dos conhecimentos de BASIC.

Aguardando logo notícias, um abraço do amigo.

PAULO ROBERTO, Niteroí, RJ.

. . . . .

µB: As suas críticas com referência a Microdigital estão expostas aqui. Caso eles queiram respondê-las, publicaremos a resposta de Microdigital na próxima edição.

Você comentou sobre uma falha na impressora. Até agora sabemos somente de dois associados que conseguiram impressoras (possivelmente são da Sinclair), embora esteja anunciada a venda da impressora desde agosto de 1982 (Cr$119.850,00) sem falar de - SOM, Conversor A-D/D-A, Modem e Diskette. Isto se chama "marketing"!

- - - : - - -

Prezado Amigo David Anderson,

Em primeiro lugar, quero parabenizá-lo pelo brilhantismo do Clube/Jornal MICRO BITS. É de excelente qualidade, e magnificamente apresentado. Meus Sinceros Parabéns!

Já que um dos pontos melhores do jornal é ajudar os seus leitores e associados, não seria possível publicar como se faz uma adaptação da função SLOW no micro NEZ8ØØØ? Você pode imaginar a profundidade que este artigo atingiria entre os possuidores (como eu) do NE?

Posso enviar muitos programas para serem publicados, mas qual seria o assunto de maior interesse?

Uma sugestão : Creio que seria interessante pedir que cada pessoa interessada em enviar um programa, anexasse a este o seu respectivo fluxograma. Assim seria bem mais fácil modificar-se o programa.

Outra coisa, porque não publicam , junto com as cartas, os endereços dos que as enviam? Qualquer dúvida, o leitor poderia entrar em contato direto com o autor do programa.

Comentário: Achei excelente a idéia de fazer um jornal para os usuários dos ZX ....., porque realmente, o manual do fabricante deixa muito a desejar.

Gostaria de obter o endereço do leitor Luiz Carlos Nardy, carta Nº 1. Interessei-me pelo que ele disse a respeito das transformações do seu NEZ8ØØØ, e como tenho um igual, queria saber de mais detalhes sobre aquelas transformações.

Planejo para breve fazer a adaptação SLOW no meu micro (pode ser que vocês publiquem como ...) e desde já gostaria saber como ela "funciona", isto é, em que pontos de um programa normal eu deveria introduzir a instrução SLOW e/ou FAST?

No manual do fabricante do NE, há uma menção da impressora, e o próprio NE já vem com as instruções LPRINT, LLIST e COPY. Será que oportunamente não poderei adaptar uma impressora no meu NE?

Estou idealizando um sistema para criar um vídeo próprio para o meu NE. Caso ele dê certo, e seja do seu interesse, posso publicá-lo no seu jornal.

ANDRÉ KOCH ZIELASKO, Av. Emancipação 402 sala 2
95590 - Tramandaí - RS.

. . . . .

µB: Agradecemos seus elogios, André!

Esperamos divulgar no próximo número de Micro Bits como adaptar o NEZ8ØØØ (e as primeiras versões do TK82C) para SLOW. Aguarde.

Não pretendemos publicar os endereços dos participantes do Clube (a não ser que seja solicitado por eles, ou se julgarmos ser de real interesse dos leitores) pois é possivel que os associados não apreciem tê-los divulgados.

- Porque usar o SLOW? Esta função é usada para jogos animados e para manter uma imagem estável na tela. Cálculos longos (por exemplo - em Xadrez) geralmente são feitos em FAST, já que é aproximadamente 6 vezes mais rápido que o SLOW.

- Impressora para o NEZ8ØØØ? Se a placa de extensão for igual ao TK82-C ou ZX81, você não deverá ter dificuldades - exceto para encontrar uma impressora!

Esperamos mais notícias sobre o seu vídeo.

- - - : - - -

Prezado amigo,

Com muita satisfação recebi o primeiro número de 'MICRO BITS', tendo mesmo me surpreendido com a sua qualidade (imagino os números futuros então!) Parabéns! .....

Dr. LÚCIO A. CASTAGNO, Pelotas, RS.

- - - : - - -

LANÇAMENTO DA MICRODIGITAL

"A Microdigital lança o TK85, com.. ..."

Bem, vocês já devem ter lido isto em todas as revistas de micro por aqui. Todos repetem quase a mesma coisa, provavelmente seguindo o "release" da Microdigital. Mas o que será que ele tem de novo? Será que é mais uma cópia dos microcomputadores da Sinclair?

Como não podemos verificar tudo em detalhes sem ter um destes TK85's na mão, só podemos tentar interpretar a (pouca) informação que foi divulgada.

Este novo modelo parece um intermediário entre os Sinclair ZX-81 e ZX Spectrum. Quer dizer, tem as funções básicas do ZX-81 (ou TK82-C) e alguma coisa do Spectrum.

Aqui está um quadro comparativo :

	TK85	SPECTRUM	TK82-C/ ZX-81
ROM	10K	16K	8K
RAM (padrão)	16K	16K	2K (TK) 1K (ZX)
RAM (opcional)	48K	48K	16-64K
VERIFY	sim	sim	não
LOAD/SAVE	4200bps	1500bps	300bps
MERGE	não (?)	sim	não
SOM	opcional	sim	não
RESOLUÇÃO GRÁFICA	44 x 64	256 x 192	44 x 64
CORES	não	8 cores (com 2 intensidades, piscante ou não)	não
IMPRESSORA	sim	sim	sim
DISKETTE	não (?)	sim	sim (?) TK não (ZX)
JOYSTICK	sim	não	sim (TK) não (ZX)
FUNÇÕES	16Ø	19Ø	151
ASCII	não	sim	não
PREÇO (*)	Cr$179.850 (16K)	Cr$86.250 (16K)	Cr$99.800 (TK - 2K)/ Cr$34.155 (ZX - 1K)

(*) usando o câmbio oficial de 27/4/83 de 1 libra esterlina = Cr$ 690,00 (aproximadamente)

Considerando os poucos dados que temos, o que podemos dizer que ele tem de bom? Além da função VERIFY (que é muito útil, òbviamente) , parece que a única outra novidade é o LOAD/ SAVE de alta velocidade.(42ØØ bps, segundo Micro Sistemas de março de 1983). Como é muito comum ter dificuldades em gravar ou carregar programas de fita cassette (já recebemos várias cartas sobre este problema), esperamos que este novo sistema realmente funcione bem. Uma coisa que não aparece na propaganda é se o TK85 tem duas velocidades de LOAD/SAVE/VERIFY. Se não tiver não terá compatibilidade entre programas gravados para os TK82-C, CP-2ØØ, etc., porque seria impossível transferir de um para o outro e vice-versa. Entretanto, Micro Mundo de 7 de fevereiro dá a impressão que tem duas velocidades - a normal e a nova "high-speed".

Uma coisa que destaca muito no quadro comparativo, são as grandes diferenças nos preços dos micros nacionais e estrangeiros. Por exemplo, o TK82-C custa 3 vezes o ZX-81, e o TK85 custa 2 vezes o ZX Spectrum, embora tenha muito menos recursos! Os fabricantes podem alegar os custos altos de importação de peças, produção muito inferior, etc., mas quase não tem custos de desenvolvimento. Só podemos esperar que eles fiquem mais em conta agora que a Prológica está entregando o CP-2ØØ, e tem mais competição no mercado.

Bem, para terminar, podemos dizer que logo que tivermos um TK85 em mãos para analisá-lo direitinho iremos informar-lhes com mais detalhes - até já !

- - - : - - -

CLUBES

Se você quiser organizar um clube no seu bairro, cidade, etc., para que participantes possam trocar idéias, programas, etc., envie-nos o seu endereço e telefone (se tiver) para ser incluído nesta coluna. Depois de organizado, podenos indicar o local de encontro, dia do mês ou semana que se reunem. Assim interessados que moram na sua vizinhança ou que estejam passando por sua cidade poderão participar.

- - - : - - -

ABAIXO A PIRATARIA !

Embora seja o segundo número de MICRO BITS, e ainda tenhamos pouca experiência sobre o mercado de software no Brasil, ficamos assustados ao saber sobre a penetração do software "pirata" que existe por aqui. Há um verdadeiro exército (talvez seja melhor chamá-lo marinha,em se tratando de piratas !) de pessoas que lucram com a venda de programas desenvolvidos por outros. Em muitos casos estas nem sequer sabem programar em BASIC; a única coisa que fazem é carregar um programa no seu micro e pronto - começam a gravar dezenas de fitas para um fácil, e imoral, lucro.

Há pouco tempo estive em São José dos Campos e tive a oportunidade de conversar com Délio Santos Lima, autor do livro "45 Programas prontos para rodar em TK82C - NEZ8000" e de vários programas em fita - Contas a Pagar, a Receber, Folha de Pagamento, jogos, etc. Ele, um dos poucos que desenvolvem seu próprio software para os TK/NE/Sinclair no Brasil, é uma das vítimas destes "marginais", e deve ter muitas cópias de seus programas espalhadas pelo país inteiro.

Entendam bem, nosso "grilo" é contra aqueles que lucram através do trabalho e sacrifício de outros. Outro assunto é a cópia de programas para dar para amigos - embora não justificável, ninguém está "faturando" e consideramos este um "mal" bem menor.

Também recebemos cartas de associados contendo programas tirados de livros e revistas. Neste caso eles deixam bem claro a fonte do programa, e não há nenhuma malícia nos seus gestos. Agradecemos este pessoal que teve a intenção de colaborar com o jornal, mas vamos deixar tudo bem claro. Não pretendemos publicar programas ou artigos copiados de qualquer fonte, a menos que tenhamos uma autorização, por escrito, do autor (ou editor da publicação). Se não for possível publicar um jornal honesto, preferimos fechá-lo desde já ! Isto não quer dizer que não aproveitaremos a idéia do programa - isto é aprendizagem - mas cópias - nunca ! Possivelmente receberemos programas para publicação que são copiados - como não temos meios de saber sobre todos os programas que foram escritos, não podemos controlar este aspecto.Neste caso, pedimos a vocês que informem a Micro Bits que já saiu aquele programa em outra revista, livro ou fita.

Voltando aos "piratas", considere estes dois perfis dos Senhores "X" e "Y":

(i) O Senhor "X", com muito esforço próprio,conseguiu aprender, através de livros importados, e outros meios, programar bem o seu microcomputador. Demorou, e o sacrifício dele e de sua família foi grande. Mas finalmente conseguiu escrever programas e começou comercializá-los. Cada programa que desenvolveu custou tempo - teve que estudar o assunto abordado no programa, verificar, modificá-lo, etc. E isto tornou-se o seu "ganha-pão", a vida dele dependendo do número de fitas ou livros que consegue vender.

(ii) O Senhor "Y", ao contrário, sai para trabalhar de 2ª a 6ª, tem um escritório confortável, tem seu 13º salário, fundo de garantia, assistência médica, férias pagas, etc., etc., e só precisa trabalhar o suficiente para não ser mandado embora. A noite, volta para casa, janta, e logo depois liga seu microcomputador, carrega uma fita que comprou(nem precisa digitar o programa na máquina!) e começa a tirar cópias a vontade. O Senhor"Y" lucra 3 vezes :-

(a) Tem um emprego garantido

(b) Aumenta sua renda com "pirataria"

(c) Não precisou gastar tempo para aprender a programar, nem desenvolver programas - só comprou uma fita, e depois copia o necessário para vender.

Agora vou perguntar para vocês - o que pensam do senhor "Y"? Não falta nada na vida dele, não é? Nada mesmo? O que falta sim é um pouco de vergonha!

Como se sentiriam os senhores "Y"s se ao final do mês, na hora de pegarem o seu salário, alguem houvesse levado 90% do valor que deveriam receber?Iriam ficar contentes?

Há poucas semanas recebemos uma longa lista de programas a venda - contendo Xadrez II, Pac-Man, Demolidor, Simulador de Vôo,Vu Kalc, Vu File, etc., etc., etc. Outros programas na lista (em BASIC) eram Apague a Trilha, Bateria Anti-Aérea, Biorritmo, Jogo da Velha, Tabuada, Invasores, etc. Pode ser mera coincidência mas estes nomes de programas, vendidos por Cr$2.000,00 cada um, também aparecem no livro de Délio, que custa Cr$3.000,00 por 45 programas!

Por enquanto, apelamos a estes "amiguinhos" que parem logo com esta atividade. Se não

souberem escrever seus próprios programas então procurem outra atividade mais honesta. Mas fiquem tranquilos, continuando assim vocês vão aprender uma lição dura.

Vocês, os nossos leitores, talvez agora compreendam porque há poucas pessoas desenvolvendo software no Brasil e porque cada programa é tão caro. Claro, por cada fita vendida há umas 20 (ou mais) cópias feitas.

O que pode ser feito para acabar com tudo isto? Muito simples. Pedimos aos nossos associados que qualquer programa que comprarem, ou virem sendo usado por colegas, que vocês suspeitam foram copiados, informe-nos com detalhes sobre o nome do programa, quem vendeu, se possui um recibo ou nota fiscal, número de cheque, etc. Mandem estas informações para nós, a/c "Projeto Limpeza", a fim de serem incluídas em nosso arquivo "negro" para futura ação!

- - - : - - -

LIVROS E OUTRAS PUBLICAÇÕES

No Micro Bits Nº 1 pedimos que quem soubesse de livrarias que vendem o livro "Mastering Machine Code on Your ZX81" nos informasse a respeito.

Luiz Carlos Nardy e Pedro Braga nos informaram que LITEC - Livraria Técnica Editora Ltda., vende este livro. (O preço em fevereiro era Cr$6.475,00). O endereço é :-
Rua dos Timbiras 257, CEP 01208, São Paulo. Tel: (011) 220-8983.

Lourival Karsten indicou uma livraria no Rio de Janeiro. É a Livraria Ciência Moderna Ltda., Caixa Postal 4420, Av. Rio Branco 158 , Rio de Janeiro, RJ. Segundo Lourival, eles prometiam a entrega dentro de mais ou menos 60 dias, e possuem um catálogo com outros livros sobre o ZX81.

Pedro Braga também indicou o endereço da revista Americana "SYNC" dedicada aos Sinclair ZX80/ZX81 :-

SYNC
CN 1986
MORRISTOWN
NJ 07960
USA

e Creative Computing, que publica os livros :-

(i) The ZX81 Companion (US$9,00)
(ii) The Gateway Guide to the ZX81 and ZX-80. (US$9,00)
(iii) Getting Acquainted with your ZX81 (US$9,00)

O endereço é :

Creative Computing
39 East Hanover Avenue
MORRIS PLAINS
NJ 07950
USA

Aqui em São Paulo encontramos livros sobre o ZX81 na :-

(i) Livraria Siciliana - Shopping Center Morumbi.
(ii) Imarés - Rua Dr. Renato Paes de Barros Nº 34, CEP 04530, São Paulo - SP.Tel: (011) 881-0200/881-1156. Falar com Neusa.

Quem tiver interesse em receber INTERFACE, a revista do 'The National ZX Users' Club for ZX80, ZX81 and ZX Spectrum', deve mandar uma libra esterlina para adquirir uma amostra da revista, ou 16 libras para assinatura por um ano para :

INTERFACE
44 - 46 Earls Court Road
LONDON W8 6EJ
INGLATERRA.

Lembre-se - Ao solicitar livros, etc., não esqueça de dizer que foi indicado pelo Clube Nacional dos TK/NE/Sinclair.

- - - : - - -

OFERTA AOS ASSOCIADOS

O livro "45 Programas Prontos Para Rodar em TK82-C e NEZ-8ØØØ" de Délio Santos Lima.

Não Associados - Cr$3.000,00 + 10% para despesas de correio.
Associados - Cr$3.100,00 (despesas incluídas)

Envie cheque nominal e cruzado, em favor de David T. Anderson para - Caixa Postal 12.464,CEP 04798 , SÃO PAULO - SP.

Válido por tempo limitado.